**#Constructur**

1. Constructor is a special type of member function because its name same as class Name.
2. Constructor have no return type.
3. Constructor is automatically called when Object is created object.

4. It is used to initialize Instance variable.

**Types of constructor's**

5.1). Default Constructor.

5.2). Parameterized Constructor.

1. **Default Constructor**  :-

Constructor with Zero Parameter is Know as defult Constructor.

1. **Parameterized Constructor** :-

A Constructor which is More then one Parameter is Know as parameterized Constructor.

**class A**

**{**

int x, y;

void show()

{

System.out.println(x);

System.out.println(y);

}

**}**

**class Q01\_Constructor**

**{**

public static void main(String args[])

{

A a = new A();

a.x = 10;

a.y = 20;

a.show();

}

**}**

Output :-

10

20

**class A**

**{**

int x, y;

void set(int x, int y)

{

this.x = x;

this.y = y;

}

void show()

{

System.out.println(x);

System.out.println(y);

}

**}**

**class Q02\_Constructor**

**{**

public static void main(String args[])

{

A a = new A();

a.set(10, 20);

a.show();

**}**

**}**

**Output :-**

**10**

**20**

**class A**

**{**

int x, y;

A(int x, int y)

{

this.x = x;

this.y = y;

}

void show()

{

System.out.println(x);

System.out.println(y);

}

**}**

**class Q03\_Constructor**

**{**

public static void main(String args[])

{

A a = new A(10, 20);

a.show();

}

**}**

Output :-

10

20

**class A**

**{**

int x, y;

A(int x, int y)

{

this.x = x;

this.y = y;

}

void show()

{

System.out.println(x);

System.out.println(y);

}

**}**

**class Q04\_Constructor**

**{**

public static void main(String args[])

{

A a = new A();

a.show();

}

}

**Output :-**

**error: constructor A in class A cannot be applied to given types;**

**A a = new A();**

**class A**

**{**

A()

{

System.out.println("Class A");

}

void show()

{

System.out.println("Show Class A");

}

**}**

**class Q05\_Constructor**

**{**

public static void main(String args[])

{

A a1 = new A();

A a2 = new A();

a1.show();

a2.show();

}

**}**

Output :-

Class A

Class A

Show Class A

Show Class A

**class A**

**{**

A()

{

System.out.println("Class A");

}

void show()

{

System.out.println("Show Class A");

}

**}**

**class Q06\_Constructor**

**{**

public static void main(String args[])

{

A a = new A();

a.A();

}

}

error: cannot find symbol

a.A();

**Constructor Chining**

**class A**

**{**

A(int x, int y)

{

System.out.println( "Sum : " + (x+y) );

}

**}**

**class B extends A**

**{**

B(int x, int y)

{

super(x, y);

System.out.println( "Sub : " + (x-y) );

}

**}**

**class Q07\_Constructor\_Chining**

**{**

public static void main(String args[])

{

B b = new B(100, 20);

}

}

**Sum : 120**

**Sub : 80**